(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 543 644 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **92310536.5**

(51) Int. Cl.[5] : **H04B 7/185**

(22) Date of filing : **18.11.92**

(30) Priority : **21.11.91 US 795610**

(43) Date of publication of application :
26.05.93 Bulletin 93/21

(84) Designated Contracting States :
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(71) Applicant : **MOTOROLA, INC.**
1303 East Algonquin Road
Schaumburg, IL 60196 (US)

(72) Inventor : **Mihm, Thomas J., Jr.**
5926 E. Fairfield
Mesa, Arizona 85205 (US)
Inventor : **Penny, Robert E., Jr.**
250 N. Corrine Circle
Gilbert, Arizona 85234 (US)

(74) Representative : **Dunlop, Hugh Christopher et al**
Motorola European Intellectual Property Operations Jays Close Viables Industrial Estate
Basingstoke, Hampshire RG22 4PD (GB)

(54) Command authentication process between a master and a slave station encrypted messages.

(57) A slave station (12), such as an orbiting satellite, and a master station (16), such as a ground control station, have their own lists of identical random pads (80,78). When the master station sends a critical command to the slave station, a selected one of the pads (86) is combined with the command and transmitted to the slave station as a data communication message (89). Each pad is used only once. The slave station evaluates the received pad value using its version of the same selected pad. If the evaluation detects correspondence, then the command is authenticated and the slave station acts upon the command. The random pads are generated (154) by the slave station. They are encrypted (160) using an asymmetric encryption process and transmitted (166) to the master station so that the master and slave stations will operate on common sets of pads.

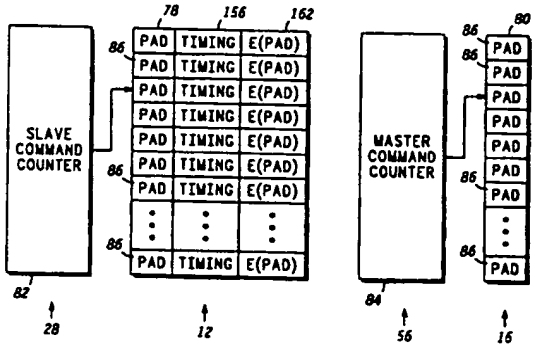


FIG. 4

EP 0 543 644 A2

Field of the Invention

The present invention relates generally to the control of one device that acts upon commands issued by another device. More specifically, the present invention relates to a process for authenticating commands issued by one device before the commands are acted upon by another device.

Background of the Invention

Many systems require a master station to issue commands for critical control functions to a remotely located slave station. One context in which this occurs is in controlling an orbiting satellite, wherein a ground-based master control station issues commands, such as orbit correction commands, to a slave satellite station. However, critical control functions are also remotely issued in connection with financial and other industrial and governmental endeavors.

The commands are considered critical due to severe harmful consequences which might result from a slave station improperly acting upon such commands. In a satellite, improper action may cause the satellite to leave its orbit or to suffer other consequences which would greatly decrease the life-span or usefulness of the satellite. Due to the great costs involved in constructing and placing a satellite in orbit, potential losses are enormous. In financial situations, money may be credited to improper accounts, again potentially leading to great financial losses. And, in other situations improper action by a slave station may pose severe risks of harm to human life and health or to the environment.

Accordingly, steps are often taken to prevent such severely undesirable results from occurring in systems where critical commands are remotely issued. Such steps are typically directed to at least two different issues. The first issue deals with insuring that a slave station will respond only to commands issued by its assigned master station. In other words, authentication processes are employed so that the slave station may have a high degree of confidence that a command it has received actually came from its assigned master station and not from some other controller. The second issue deals with insuring that a slave station which receives a command from its assigned master station has in fact received the intended command and not some other command. In other words, confirmation processes are employed to insure that no error has occurred in communicating a command.

Numerous efficient, effective, and otherwise satisfactory confirmation processes are known to those skilled in the art. On the other hand, conventional authentication processes fail to provide satisfactory solutions to authentication needs of systems such as those described above. Conventional authentication processes are complicated to implement and to operate. Such conventional processes often include dedicated "secure" hardware for implementing encryption and decryption processes. While such conventional processes do achieve effective levels of authentication security, the secure hardware's weight, cost, and limited reliability, when compared to a process using no additional hardware, are undesirable features of the conventional processes.

In addition, many conventional processes utilize a symmetric encryption process. Symmetric encryption processes use a single key for encryption and decryption. This key is kept secret so that only the master and slave stations know it. However, the system must employ a re-keying scheme in case a breach of security causes the secret key to become known or some failure keeps authorized messages from being authenticated. Such re-keying schemes become extremely complicated and costly in order to maintain a high level of authentication security.

While the above problems plague all conventional authentication processes, the problems become especially troublesome in connection with the control of satellites. In a satellite control system, physical access to a satellite after the satellite is in orbit is extremely impractical. In addition, the penalties paid for increased weight needed for secure hardware and for reduced overall reliability from using secure hardware are amplified.

Summary of the Invention

In accordance with a first aspect of the present invention there is provided a method for communicating commands to a slave station so that the commands can be authenticated by the slave station. In accordance with this method of the present invention, a pad is received from the slave station. The pad is saved at least until a need arises for communicating a command to the slave station. This pad is combined with the command to form a message, and the message is transmitted to the slave station.

In accordance with a second aspect of the present invention there is provided a method for authenticating commands received from a master station prior to carrying out instructions communicated by the commands. In accordance with this method of the present invention, a pad is generated using a random generating process. This pad is saved and transmitted to the master station. Later, a data communication message is received. A command and a pad value are recovered from the message. The recovered pad value is compared with the saved pad to detect correspondence. If the correspondence is not found, the command is not acted upon.

Accordingly, the present invention provides an improved authentication process.

Another advantage of the present invention is

that an authentication process which does not require the use of hardware dedicated to the process is provided.

Yet another advantage is that the present invention provides an authentication process which is simple to implement and to maintain.

Another advantage is that the present invention provides a high level of authentication security.

Brief Description of the Drawings

A more complete understanding of the present invention may be derived by referring, by way of example only, to the detailed description and claims when considered in connection with the Figures, wherein like reference numbers refer to similar items throughout the Figures, and:

FIG. 1 shows a satellite-based communication network which encircles the earth and within which a preferred embodiment of the present invention operates;

FIG. 2 shows a block diagram of hardware utilized by a satellite slave station and a ground control master station in accordance with a preferred embodiment of the present invention;

FIGs. 3A-3B together show a flow chart of tasks performed by the master station and the slave station in communicating an authenticated command therebetween in accordance with a preferred embodiment of the present invention;

FIG. 4 shows a block diagram of pad-related data structures utilized by the master and slave stations in accordance with a preferred embodiment of the present invention;

FIGs. 5A-5B show a block diagram of a command structure and its relationship to a data communication message in accordance with a preferred embodiment of the present invention; and

FIGs. 6A-6B together show a flow chart of tasks performed by the master station and the slave station in establishing lists of pads in accordance with a preferred embodiment of the present invention.

Description of the Preferred Embodiment

FIG. 1 illustrates a satellite-based communication network 10. Network 10 is dispersed over the entire world through the use of many above-the-earth vehicles, such as orbiting satellites 12. In the preferred embodiment, satellites 12 occupy polar, low-earth orbits, which cause them to move relative to the earth in south-to-north and north-to-south directions. Together, satellites 12 form a constellation in which satellites 12 remain relatively stationary with respect to one another, except for their orbits converging and crossing over each other in the polar regions.

Satellites 12 communicate with devices on the ground through many gateways 14, of which FIG. 1 shows only one, a few ground control stations (GCS's) 16, of which FIG. 1 shows two, and any number of radiocommunication units 18, of which one is shown in FIG. 1. Radiocommunication units 18 may be located anywhere on the face of the earth. Gateways 14 are preferably distributed over the surface of the earth in accordance with geo-political boundaries. GCS's 16 preferably reside in the northern latitudes, where the convergence of the orbits of satellites 12 causes a greater number of satellites 12 to come within direct line-of-sight view of a single point on the surface of the earth with respect to more equatorial latitudes. Preferably, around four GCS's are used so that all satellites 12 in the constellation may at some point in their orbits come within direct view of their assigned GCS 12.

Nothing prevents gateways 14 and GCS's 16 from being located together on the ground. However, gateways 14 serve a different function from that of GCS's 16. Preferably, gateways 14 primarily operate as a communications nodes 20 in network 10. Diverse terrestrial-based communications systems, such as the worldwide public switched telecommunications network (PSTN), access network 10 through gateways 14. Network 10 may establish a communication circuit through a gateway 14 and the constellation of satellites 12 to a particular one of satellites 12 which is then-currently over a desired location for the other side of the circuit, as illustrated at communication path 22 between a satellite 12 and radiocommunication unit 18. Thus, each of satellites 12 also serves as a node 20 of network 10.

GCS's 16 perform telemetry, tracking, and commanding (TT&C) functions for the constellation of satellites 12. Preferably, extensive security measures are employed at each of GCS's 16 to prevent unauthorized control of satellites 12. GCS's 16 also serve as nodes 20 for network 10. However, GCS's 16 differ from gateways 14 in that they primarily perform TT&C functions for network 10 rather than interface other communication networks and systems to network 10. Due to the northern latitudes at which GCS's 16 are located, one or more satellites 12 are always in view of each GCS 16. A GCS 16 may communicate with its assigned satellites 12 within network 10 directly, as shown at direct links 24, when their orbits bring such assigned satellites 12 over the GCS 16. In addition, any GCS 16 may communicate with any satellite 12 through an overhead satellite 12 and various intermediate nodes 20 of network 10.

In the preferred embodiment, direct links 24 allow GCS's 16 to communicate with their assigned satellites 12 over such high frequencies, preferably in the Ka band, that only substantially line-of-sight propagation of communication signals results. An unauthorized party must be near a direct line-of-sight between a satellite 12 and a GCS 16 to receive such commu-

nication signals. Thus, the unauthorized receiving of these signals is difficult to accomplish for communication signals transmitted from the ground toward a satellite 12 because it requires a receiver physically placed between the ground-based transmitter and the space-based satellite receiver.

In accordance with their control function, GCS's 16 transmit commands to their assigned satellites 12. These commands are communicated to satellites 12 in data communication messages. Such commands control numerous diverse operational aspects of satellites 12. For example, such commands may activate thrusters to affect orbits, energize power systems to affect a satellite's ability to serve as a node in network 10, and control switching systems to influence a satellite's capacity to handle communication traffic. These and other commands are of critical importance to the effective operation of network 10. Alternatively, a critical command may be a NO-OP or a security trap. Consequently, network 10 takes steps to insure that these and other critical commands are authenticated to prevent damage which might otherwise be caused by mischievous tampering, negligent procedures, or deliberate sabotage.

FIG. 2 shows a block diagram of hardware 16' used by a master station, such as a GCS 16, and hardware 12' used by a slave station, such as a satellite 12. As is common for master and slave stations, stations 12 and 16 represent computer controlled devices. In particular, slave station hardware 12' includes a processor 26 which communicates with a memory 28. Memory 28 includes a permanent memory 30, whose programming cannot be altered by processor 26, and a temporary memory 32, whose programming can be altered by processor 26. Those skilled in the art will recognize that the many procedures, processes, and tasks performed by slave station 12 are controlled by processor 26 in response to programming instructions stored in memory 28.

In particular, under the control of processor 26 various actuators 34 are activated and deactivated. Such actuators may cause thrusters to fire, deploy and alter positions of solar panels, and control antenna movement, to name a few activities. Likewise, various switches 36 may be operated under the control of processor 26. Switches 36 may, among other things, control the application of power to various sub-systems included within slave station 12. Furthermore, various communication controls 38 are manipulated under the control of processor 26. Controls 38 are used to allow slave station 12 to serve as a communications node in the operation of network 10.

Processor 26 also communicates with a timer 40, which aids processor 26 in maintaining a date and time-of-day clock. Processor 26 transmits data, preferably using conventional QPSK techniques, by supplying such data to a modulator 42. Processor 26 also couples to a synthesizer 44 to control frequency channels over which the data are transmitted. After modulating the data in modulator 42, a communication signal is transmitted from slave station 12 at a transmit section 46 of an antenna 48. Processor 26 receives data via a receive section 50 of antenna 48, which couples to a demodulator 52. Demodulator 52 has a control input which couples to an output of synthesizer 44. Demodulator 52 demodulates the received data, which are then routed to processor 26.

Control station hardware 16' includes similar circuits. In particular, a processor 54 couples to and communicates with a memory 56. The many procedures, processes, and tasks performed by master station 16 are controlled by processor 54 in response to programming instructions stored in memory 56. Processor 54 additionally couples to various input/output (I/O) devices 58. I/O devices 58 include any conventional keyboard, pointing device, video display terminal, printer, and other automated systems, such as other computers or workstations coupled through a computer network or direct data communication link.

Processor 54 also communicates with a timer 60, which aids processor 54 in maintaining a date and time-of-day clock. Processor 54 transmits data, preferably using conventional QPSK techniques, by supplying such data to a modulator 62. Processor 54 also couples to a synthesizer 64 to control frequency channels over which the data are transmitted. After modulating the data in modulator 62, a communication signal is transmitted from master station 16 at a transmit section 66 of an antenna 68. Processor 54 receives data via a receive section 70 of antenna 68, which couples to a demodulator 72. Demodulator 72 has a control input which couples to an output of synthesizer 64. Demodulator 72 demodulates the received data, which are then routed to processor 54. As discussed above in connection with FIG. 1, the data may be communicated through a direct link 24 or via network 10 through one or more intermediate nodes 20.

Accordingly, slave station 12 and master station 16 include the type of conventional circuits which are required to operate virtually any computer controlled radiocommunication apparatus. In particular, stations 12 and 16 do not include circuits which are dedicated to the performance of command authentication. Rather, command authentication is performed primarily within processors 26 and 54 in accordance with programming instructions stored in memories 28 and 56, respectively. Processors 26 and 54 are the same processors used to operate stations 12 and 16, respectively. Thus, hardware reliability for command authentication is roughly equivalent to the reliability of the corresponding processors, memories, and radio communication support circuits. Moreover, the overall reliability of stations 12 and 16 is not reduced, nor is the weight of satellite slave station 12 in-

creased, by the inclusion of command authentication hardware.

Of course, those skilled in the art will appreciate that the hardware shown in FIG. 2 may be substantially altered without causing a substantial change in the functions performed. For example, various timing, switching, and communication control functions may be included within processor 26, and processors 26 and 54 may be implemented using either a single processor circuit or multiple processor circuits. Likewise, modulation and demodulation functions may be configured in diverse ways, including the addition of intermediate modulation stages. Moreover, those skilled in the art will realize that stations 12 and 16 may utilize a common antenna for both the transmission and reception of communication signals.

FIGs. 3A-3B together show a flow chart of tasks performed by master station 16 and slave station 12 and show the relationships between such tasks. The particular tasks shown in FIGs. 3A-3B relate to the sending and authentication of a command from master station 16 to slave station 12. Moreover, the particular tasks shown in FIGs. 3A-3B take place during the normal operation of stations 16 and 12. In other words, processors 26 and 54 (see FIG. 2) perform the tasks shown in FIGs. 3A-3B while also attending to the normal operations of stations 12 and 16, respectively.

With reference to FIG. 3A, from time-to-time master station 16 receives a slave command count data communication message from one of its assigned slave stations 12, as indicated in a task 74. Preferably, slave stations 12 incorporate the slave command count value along with other routine, operational telemetry that is communicated to master station 16, as shown in a task 76. Thus, master station 16 receives the slave command count data regularly. Moreover, this data message may be sent in the clear (i.e. need not be encrypted).

FIG. 4 shows a block diagram of pad lists 78 and 80, formed in memories 28 and 56 of slave and master stations 12 and 16, respectively, for use in the command authentication process of the present invention. Pad lists 78-80 operate in connection with slave and master command counters 82-84, respectively. Each of pad lists 78-80 includes many one-time pads 86, the precise number of which is not important to the operation of the present invention.

The term "one-time pad" refers to a stream of random information of a finite length that is combined, for example, through a linear process such as addition with a message of the same length as a means of encrypting the message. Decryption of the message requires the use of the same stream of random information or pad combined in the same or a complementary way. It is "one-time" because the same random stream is substantially never used for more than one message. Thus, a one-time pad process is said to be unconditionally secure. Each pad 86 represents a variable in which a pad value is stored. The term "pad" refers both to the variable and to the value itself. Pads 86 are generated through conventional random number generation processes. The generation of pads is discussed in more detail below. Accordingly, each of lists 78 and 80 includes a set of random numbers. Moreover, lists 78 and 80 include the same set of random numbers, preferably in the same order.

Counters 82-84 serve as pointers to lists 78-80, respectively. When counters 82 and 84 have the same value, they identify the same pad without actually mentioning the identified pad's value. Thus, in task 76 (see FIG. 3A) slave station 12 identifies a particular pad 86 to master station 16 without actually mentioning the pad's value. For this reason, the slave command count data message transmitted and received at tasks 76 and 74, respectively, may be sent in the clear.

With reference back to FIG. 3A, when master station 16 receives the slave command count value in task 74, it synchronizes its master command count counter 84 (see FIG. 4) to this value. Pads 86 are utilized by the preferred embodiment of the present invention in a manner to be discussed in more detail below. For reasons which will become apparent, master station 16 and slave station 12 utilize a common pad 86 for each command to be authenticated in accordance with the preferred embodiment of the present invention. In the event that master and slave command counters 84 and 82 might become unsynchronized, master station 16 and slave station 12 would then attempt command authentication using diverse pads 86. As a result, authorized commands would then not be authenticated. Normal operation of network 10 should keep counters 82-84 in synchronization. Nevertheless, should any synchronization problem between counters 82 and 84 occur, master and slave stations 16 and 12 will soon re-synchronize due to the performance of tasks 74-76.

While attending to normal operations, master station 16 may receive a data communication message that acknowledges the receipt of a command by a slave station 12, as shown at a task 87. Since master station 16 has not sent a command at this point in the process, such an acknowledgement is not expected and indicates a problem. For example, such an acknowledgement might indicate an attempt at tampering with the slave station 12 or a failure within slave station 12. In any event, when task 87 encounters the acknowledgement message, an appropriate warning message may be activated so that corrective action can be taken. The warning message may be displayed, printed, audibly indicated, or otherwise brought to the attention of a human operator through appropriate control of I/O devices 58 (see FIG. 2).

While attending to normal operations, slave station 12 may need to act upon or otherwise carry out

commands which it has previously received, as shown at a task 88. Whether such commands require action depends upon the contents of temporary memory 32 (see FIG. 2).

FIG. 5A-5B respectively show a block diagram of a data message 89 sent to slave station 12 and a command data structure 90 which may be communicated by the data message. Data structure 90 may advantageously be repeated within memory 32 for any number of commands. Structure 90 includes a time stamp which instructs processor 26 when to carry out the command's instruction. An enablement data element is used to communicate whether the corresponding command is enabled. A disabled command may not be acted upon, but an enabled command may be acted upon. A command identifier data element communicates precisely what action slave station 12 should take, and various parameters characterize that action. For example, the command identifier may instruct the firing of a particular thruster, and the parameters establish the duration of the firing. The time stamp determines when the thruster firing should begin.

With reference back to FIG. 3A, during task 88 processor 26 (see FIG. 2) routinely monitors temporary memory 32 (see FIG. 2) for enabled commands which need processing at the then-current time. When the enablement and timing requirements are met, processor 26 carries out the command. Although not specifically shown, slave station 12 may then transmit a message to master station 16 to communicate the fact that the command has been acted upon.

During the normal operation of master station 16, a need eventually arises which requires that a command be sent to a particular slave station 12. For example, an orbit correction may be required or particular data may be needed from a slave station 12. When this need arises, master station 16 formulates the appropriate command, as indicated at a task 92. This command preferably takes the form illustrated in data structure 90 of FIGs. 5A-5B.

With reference to FIG. 5A, the need may, but need not, require the execution of several commands, as illustrated in message 89. In formulating the commands, addresses are associated with the data that serve as command structures 90. These addresses indicate the locations within temporary memory 32 (see FIG. 2) where the commands are to be stored. Further, an appropriate header is associated with the message. The header essentially identifies the slave station 12 to which the commands are directed and instructs that slave station 12 to load the command data within its temporary memory 32 at the indicated addresses.

With reference back to FIG. 3A, after task 92, an optional task 94 may be performed. In the preferred embodiment of the present invention a command may be classified as either a critical command or a non-critical command. Critical commands require authentication due to severe consequences which might possibly result from inappropriately performing such commands. On the other hand, non-critical commands do not have such severe consequences if inappropriately performed. One example of a non-critical command is an instruction to adjust satellite transmitter power within predetermined limits. Another example is an instruction to send a current command count. When a non-critical command is to be transmitted to a slave station 12, no authentication is required, and task 94 is not performed.

For critical commands, task 94 gets a pad 86 (see FIG. 4), encrypts an appropriate authorization code (AC) with the pad 86, and combines the encrypted AC with the commands in message 89 (see FIG. 5). The pad 86 obtained in task 94 is preferably the pad 86 indicated by master command counter 84 (see FIG. 4). In the preferred embodiment, this is simply the next pad 86 occurring within list 80 (see FIG. 4) after the previously used pad 86. The encryption performed by task 94 is preferably a conventional linear process, such as a bit-by-bit Exclusive Or operation. In the preferred embodiment, pads 86 and AC's have the same number of bits. Since pad 86 is a random number, this scrambles the AC. As shown in FIG. 5, the encrypted AC is appended to and becomes part of message 89.

One advantage of allowing the distinction between critical and non-critical commands is that the non-critical commands may be issued from other locations than a GCS 16 (see FIG. 1). For example, gateways 14 (see FIG. 1) may exert a limited amount of control over satellites 12 (see FIG. 1). Security is not compromised, and the complexity of the authentication process is minimized by refraining from transporting and synchronizing pad lists 80 (see FIG. 4) between all locations which may advantageously exert a limited amount of control over satellite 12. Rather, pad list 80 is kept within the secure confines of GCS 16. Another advantage of allowing the distinction between critical and non-critical commands is that a slave station 12 may be instructed to send its current command count to speed any resynchronization process.

After task 94, or task 92 for non-critical commands, a task 96 transmits data communication message 89 (see FIG. 5A), containing the commands and optionally containing the encrypted AC, to the target slave station 12. Master station 16 has now temporarily completed its part of the authentication process, and the target slave station 12 continues the authentication process.

In particular, while performing normal operations slave station 12 eventually receives data communication message 89, as indicated at a task 98. Task 98 then recovers the command portions of message 89.

In a query task 100, slave station 12 examines the command identifier portions of the commands in message 89 to determine if at least one command in message 89 is a critical command. Slave station 12 may decide whether a command is critical or not by performing a table look-up operation upon a table (not shown), preferably programmed into permanent memory 30 (see FIG. 2).

If no command included within message 89 is a critical command, then slave station 12 performs a task 102. In task 102, processor 26 of slave station 12 formulates an acknowledgement message which carries data indicating that a non-critical command has been received. Typically, a non-critical command will be programmed with its enablement data element (see FIG. 5) in an enabled state, and slave station 12 will act upon it in due course. The time stamp portion of the command (see FIG. 5) may be programmed to indicate that the command should be performed immediately. In this case, the next time slave station 12 performs task 88, the command will be acted upon.

If at least one command included within message 89 is a critical command, station 12 performs a task 104. Task 104 obtains a pad 86 from list 78 (see FIG. 4). In the preferred embodiment of the present invention, the particular one of pads 86 selected in task 104 corresponds to the value then-currently in slave command counter 82 (see FIG. 4). This pad is preferably the next pad 86 occurring within list 78 after the previously used pad 86.

Task 104 then uses the selected pad 86 to recover a pad value from the encrypted version of the authorization code (AC) included in message 89 (see FIG. 5A). As discussed above, in the preferred embodiment of the present invention a pad 86 is used to encrypt an AC using an Exclusive Or operation. Task 104 decrypts using an Exclusive Or operation between the selected pad 86 and the encrypted AC.

Referring now to FIG. 3B, processor 26 of slave station 12 next performs a query task 106. Task 106 evaluates the pad value results of task 104 to determine whether the AC is a correct AC. This determination may be made by comparing the recovered AC with one or more values programmed in memory 28, and preferably permanent memory 30 (see FIG. 2). If the recovered AC is correct, then the recovered pad value matched the selected pad 86, and a correspondence between the pad 86 used to encrypt the AC and the pad 86 used to decrypt the encrypted AC has been established. In this situation, the command may be considered authenticated. Since pads 86 are substantially random numbers, slave station 12 can have a level of confidence roughly equivalent to $1 - 2^{-n}$, where n is the number of bits included in each pad 86, that the command originated from its master station 16.

In a first alternative embodiment, the AC need not be used at all in the authentication process. Rather than encrypting and decrypting an AC, a pad 86 may be included directly within message 89 by master station 16 and directly recovered from message 89 by slave station 12. In this situation slave station 12 may evaluate the recovered pad value in a more straightforward fashion. In particular, the recovered pad value may be directly compared with the selected pad from list 78 to detect correspondence.

In a second alternative embodiment, the AC may advantageously be encrypted with a pad 86 as discussed above. However, slave station 12 may decrypt the encrypted authorization code by an Exclusive Or operation with an approved AC rather than with the selected pad. In this situation, slave station 12 evaluates the recovered pad value by directly comparing the results of the Exclusive Or operation with the selected pad 86. Regardless of the specific embodiment utilized, those skilled in the art will appreciate that slave station 12 evaluates a recovered pad value to detect a correspondence between the pad 86 used by master station 16 and the pad 86 selected by slave station 12.

When task 106 detects pad correspondence, a task 108 formulates an "Authenticated Command" acknowledgment message. A task 110 then increments slave command counter 82 (see FIG. 4). The incrementing of counter 82 signifies that the previously selected pad 86, which was utilized above in tasks 104-106, will not be used again by the authentication process. Of course, those skilled in the art will appreciate that this does not mean that the pad value carried by this previously selected pad 86 cannot occur again. Rather, the probability of any pad 86 having any particular value will remain approximately 2^{-n}.

When task 106 fails to detect pad correspondence, a task 112 disables the command. The command may be disabled by manipulating the enablement data element in command data structure 90 (see FIG. 5). As a result of disabling the command, satellite 12 will not act upon the command. Next, a task 114 formulates a "failed authentication" acknowledgement message. This "failed authentication" message may advantageously include details relating to the identity of the command or may include the entire command. Such data may help master station 16 to determine the cause of the failed authentication acknowledgement message and to take appropriate responsive action.

After the acknowledgement message of task 102, 110, or 114 has been formulated, a task 116 transmits the acknowledgement message back to master station 16. At this point, slave station 12 has completed its part in the authentication process, and the procedure illustrated in the flow chart of FIGs. 3A-3B repeats. In other words, slave station 12 continues with normal slave station operations. Such normal operations will cause slave station 12 to act upon enabled commands at the programmed times, as dis-

cussed above in connection with task 88.

While slave station 12 has been authenticating message 89 (see FIG. 5A), master station 16 has been engaging in normal station operations. Eventually, the acknowledgement message is received by master station 16, as indicated at a task 118. A query task 120 evaluates the acknowledgement message. When an Authenticated Command acknowledgement message is received, a task 122 increments master command counter 84 (see FIG. 4). The incrementing of counter 84 signifies that the previously selected pad 86, which was utilized above in task 94, will not be used again by the authentication process. The incrementing of counter 84 also keeps counter 84 synchronized with slave command counter 82 (see FIG. 4). The next critical command will utilize a different pad 86 for authentication purposes. By refraining from using the same pad twice, the security level of the authorization process is maintained at a high level.

After task 122, a task 124 performs any confirmation processes which may be required. As discussed above, confirmation processes differ from authentication processes. Authentication processes insure that commands originate only from authorized sources. Confirmation processes insure that the commands received by slave station 12 are the intended commands.

Due to the programmable nature of the present invention, great flexibility in confirmation processes is provided. These confirmation processes are controlled by master station 16. For non-critical commands, no confirmation is needed. Rather, non-critical commands do not lead to severe consequences if improperly performed, and such commands may simply be repeated as necessary.

For critical commands, procedures adopted by master station 16 may advantageously reflect how critical different commands are. For example, low-level critical commands need not require any confirmation. Such commands may be sent with a time stamp and enablement data element of command structure 90 (see FIG. 5B) set for desired execution by slave station 12. For low-level critical commands master station 16 need not take any action in task 124. Medium-level critical commands may be sent with a time stamp and enablement data element of command structure 90 set for a future execution by slave station 12. After acknowledgement is received, master station 16 may confirm the command by having slave station 12 read the command back to master station 16 in task 124. If the command is correct on the read-back operation, master station 16 takes no further action. But, if the read-back operation indicates an incorrect command, then master station 16 may disable or write over the command. High-level critical commands may be sent in a disabled state. A read-back operation may be performed in task 124. If the read-back is correct, then another command message may enable the command, and another read-back operation may confirm that the command was in fact enabled. If high-level critical commands are incorrect in the read-back operation, chances are that no preventive action is needed because the command should be disabled anyway. These and other confirmation processes which are known or obvious to those skilled in the art are contemplated at task 124.

After task 124 or when task 120 detects an non-critical command acknowledgement, the procedure illustrated in the flow chart of FIGs. 3A-3B repeats for master station 16. In other words, master station 16 continues with normal master station operations.

When task 120 detects a failed authentication acknowledgement message, a query task 126 determines whether to exit a programming loop. If master station 16 decides not to exit the loop, then program control loops back to task 94, discussed above. Generally speaking, task 94 appends a pad to a critical command and a subsequent task sends the command and pad to slave station 12. However, for this iteration of the loop, master station 16 assumes that slave command counter 82 and master command counter 84 (see FIG. 4) have gotten slightly out of sync. Accordingly, the command is repeated using subsequent pads 86 from list 80 (see FIG. 4) in an attempt to get slave station 12 to authenticate the command. The command transmission is repeated with different pads 86 for a predetermined number of iterations or until an Authenticated Command acknowledgement message is received. If this predetermined number is reached, task 126 exits the loop to a task 128, which activates an appropriate warning message. After task 128, program control for master station 16 repeats the procedure illustrated in the flow charts of FIGs. 3A-3B.

The procedure discussed above in connection with FIGs. 3A-3B assumes that lists or sets of pads 86 are already stored within memories 32 and 56 of slave and master stations 12 and 16, respectively. In the preferred embodiment of the present invention, no limits are placed on the number of critical commands which may be transmitted between stations 16 and 12. In other words, the procedures shown in FIGs. 3A-3B repeat for an indefinite number of iterations. The number of iterations is potentially extremely large over the life span of slave station 12. In addition, a finite amount of memory is dedicated to storing lists 78 and 80. Accordingly, the present invention includes a procedure for creating lists 78 and 80 as needed. This set-up procedure is shown in a flow chart presented in FIGs. 6A-6B.

The procedure for creating lists 78 and 80 utilizes a public key or asymmetric encryption system. Those skilled in the art will appreciate that an asymmetric system uses one key or value to encrypt messages

and another key to decrypt messages. The encryption key is considered a public key because it can be made public without compromising system security.

With reference to FIG. 6A, an initial secret, decryption key and an initial set of pads 86 are stored within memory 56 of master station 16 at a task 130. These initial values are generated using conventional processes before slave station 12 is installed, or launched in the case of a satellite slave station 12. The same initial set of pads 86 is stored within memory 28 of slave station 12 at a task 132. Task 132 also stores an initial public encryption key in memory 28 of slave station 12. The public key is preferably stored in the permanent portion 30 of memory 28 so that, with a high degree of confidence, it will remain available throughout the life of slave station 12. Initial pads 86 may be stored in either permanent memory 30 or temporary memory 32 of slave station 12.

In FIG 6A, task 132 occurs before installation of slave station 12 at a remote location (e.g. in orbit), so physical access to slave station 12 exists prior to launch. If there is concern about physical security of the initial set of pads 86 then the initial set of pads may be generated by slave station 12 after installation and automatically sent to master station 16 using substantially the process shown in FIG. 6B for subsequent pads.

After task 132, a task 134 calls for launching or otherwise installing slave station 12 at a remote location. In accordance with the process of the present invention, future physical access to slave station 12 is not needed. Immediately after slave station 12 has been installed, an urgent need may arise for issuing some initial commands. Master station 16 transmits such initial commands at a task 136 and slave station 12 receives such initial commands at a task 138. The authentication process discussed above in connection with FIGs. 3A-3B is used in tasks 136-138. The commands are covered using the initial pads 86 loaded prior to launch and saved in tasks 130 and 132, or alternatively initial pads 86 generated in orbit via tasks 154-170. Slave station 12 acts upon such initial commands, when authenticated, as discussed above. In other words, tasks 136-138 relate to the normal operation of master and slave stations 16 and 12, respectively.

In accordance with this normal operation, slave station 12 performs a query task 140. Task 140 determines whether a new set of pads 86 is needed. This need may be precipitated by having already consumed most of pads 86 within list 78 (see FIG. 4). Alternatively, task 140 may reach this conclusion from a failure to receive any authenticated critical command from master station 16 within a predetermined period of time. Such a lack of critical commands over a long period possibly indicates that a serious failure in the authentication system has occurred. A new set of pads will often permit recovery from such serious failures.

When task 140 decides that a new set of pads is not needed, slave station 12 continues with normal operations. In the meantime, after launching slave station 12 master station 16 includes a query task 142 in its normal operations. Task 142 determines whether slave station 12 is overhead and within a predetermined timing window. Slave station 12 is directly overhead any GCS 16 specifiable by a master GCS 16 when master station 16 can communicate with it through direct link 24 (see FIG. 1). The predetermined timing window is designed to occur only when slave station 12 is overhead. So long as the location and timing of slave station 12 do not meet the criteria of task 142, master station 16 continues with its normal operations. On the other hand, when slave station 12 is overhead and within the predetermined timing window, master station 16 performs a task 144.

Task 144 transmits a new public key to slave station 12 in a set-up data communication message. This new public key is not encrypted, and need not be decrypted at slave station 12. The set-up message is formatted as a command to slave station 12 and is considered a critical command. Thus, the set-up message includes an authentication code. Immediately after launching slave station 12, this authentication code relies upon one of the initial pads 86 saved above in task 130 (or 170). As discussed above in connection with FIGs. 3A-3B, master station 16 may expect some sort of acknowledgement message in response to the set-up message.

Slave station 12 receives the set-up message at a task 146. The new public key is recovered from the message, the message is authenticated, and an acknowledgement message is transmitted back to master station 16. After task 146, a query task 148 then determines whether satellite 12 is over master station 16 and within the predetermined timing window. The location data may be obtained by determining whether the set-up message was received over direct link 24 (see FIG. 1) or through other links in network 10. The timing window data may be programmed in permanent memory 30. If task 146 fails to determine that slave station 12 is at the prescribed location and time, the newly received public key is discarded in a task 150, and slave station 12 continues with its normal operations. On the other hand, if the location and timing window are correct, then the new public key is saved within temporary memory 32 in a task 152.

Security is enhanced by limiting the transmission of new public keys to occurring only while slave station 12 is overhead. In the preferred embodiment, slave stations 12 are overhead 2-14 times daily, depending on latitude. As discussed above, the communication signals transmitted from the ground to a satellite are extremely difficult to intercept, and the chances of mischief resulting from an unwanted interception of the new public key are greatly reduced.

Referring to FIG. 6B, after task 152 and when task 140 determines that a new pad list needs to be generated, a task 154 generates and saves a new set of random pads 86 in list 78 (see FIG. 4). Task 154 also generates and saves a set of random timing values, and saves them in a list 156 (see FIG. 4) associated with the random pads 86.

Task 154 need not be performed as a high priority task by slave station 12. Rather, slave station 12 may perform more pressing normal operations and eventually generate the random values as it has time to do so. Those skilled in the art will appreciate that pseudorandom number generation processes are well known and that such processes may advantageously be seeded with arbitrary values, such as time values or other miscellaneous but time-variant data, to produce substantially random values. The random timing values are formatted as durations, with some predetermined limitation on a maximum duration allowable. These timing values will be used in a subsequent task for transmitting the random pads to master station 16.

After task 154, slave station 12 performs a task 160 to encrypt the random pads generated above in task 154 using the highest priority public key available to slave station 12. The preferred embodiment of the present invention utilizes a well known encryption technique, such as the RSA or Hellman techniques. The encrypted pads (E(Pad)) are stored in a list 162 (see FIG. 4), in which the encrypted pads are associated with the pads 86 and random timing values of lists 78 and 156, respectively. Task 160 may also be performed in a low priority mode by slave station 12, where slave station 12 is free to perform more pressing tasks before completing task 160. Accordingly, encryption demands do not adversely consume critical processing power needed for the normal operation of slave station 12.

The highest priority public key available to slave station 12 is usually the one most recently received from master station 16. Thus, the public key initially loaded in task 132 need not be used unless no other public key has been received from master station 16. In addition, when task 140, discussed above, decides that a new set of pads is needed because no critical messages have been received from master station 16 within a predetermined period of time, task 160 may utilize the initial public key for encryption as a default under the assumption that the other public key may have been corrupted.

After task 160, slave station 12 again attends to its normal operations. These normal operations include a query task 164, which examines list 156 of random timing durations. If the duration indicated in association with the next pad 86 has transpired, then a task 166 transmits the corresponding encrypted pad from list 162 to master station 16. This pad is included in a data communication message with a count value from slave station command counter 82. In an alternate embodiment, task 166 transmits several encrypted pads, but less than the entire set of pads, to master station 16. After task 166, slave station 12 returns to its normal operations, including the performance of task 164. Slave station 12 will loop through tasks 164 and 166 until all pads have been transmitted to master station 16. The pads will have been transmitted at random intervals to further reduce the likelihood of them being intercepted and decrypted. This transmission of encrypted pads is not restricted to occurring only while slave station 12 is over its master station 16. Rather, the data messages which carry encrypted pads may be transmitted from slave station 12 through network 10 (see FIG. 1) at any time and from any location.

While slave station 12 has been generating random numbers and encrypting pads, master station 16 has continued with its normal operations. Eventually, master station 16 receives a data communication message which carries encrypted pads and the slave command counter value, as indicated at a task 168. When this message is received, a task 170 decrypts the received pad(s) using a secret key. This secret key corresponds to the last public key sent to slave station 12 or to the initial public key loaded in slave station 12. Task 170 saves the decrypted pads in list 80 (see FIG. 4) and also saves the slave command count value for use in synchronization. After task 170, master station 16 continues with its normal operations, which, from time-to-time, includes the receipt of additional encrypted pads in task 168.

Accordingly, random pads are generated as necessary, largely under the control of slave station 12. However, master station 16 may repeat the transmission of new public keys to slave station 12 on a regular interval, when a compromise in security is suspected or when an authentication process failure is suspected. Slave station 12 will respond to this new public key with a new set of pads.

In summary, the present invention provides an improved authentication process. The process of the present invention does not require the use of hardware dedicated to authenticating commands. Rather, the process of the present invention is simple to implement and to maintain. No physical access to a slave station is required, and the process may be implemented along with other normal operational processes using only hardware which is needed to perform such normal operations.

In addition, the present invention provides a high level of authentication security. One-time pads are generated as needed for authenticating commands, then discarded. The same pad is not reused, and the process is considered unconditionally secure. The pads are communicated between the slave and master stations using an asymmetric, public key encryption system. These pads are not communicated often, and the pads are communicated at random intervals

to further enhance security. Re-keying is a simple operation due to the use of a public key in the slave stations. Security is further enhanced by restricting the re-keying operation to occurring only when a satellite slave station is positioned in a particular location and timing window. The particular location is one where unintended receipt of communication signals transmitted from the master station to the slave station is exceptionally difficult due to the line-of-sight nature of the signals.

The present invention has been described above with reference to preferred embodiments. However, those skilled in the art will recognize that changes and modifications may be made in these preferred embodiments without departing from the scope of the present invention. For example, those skilled in the art will appreciate that the numbers described herein as being random are to be considered substantially random for the practical purposes with which the present invention is concerned, but may in an absolute sense be considered pseudorandom numbers. Moreover, those skilled in the art will appreciate that the hardware and specific tasks discussed herein are subject to substantial modifications while still accomplishing substantially the same results. These and other changes and modifications which are obvious to those skilled in the art are intended to be included within the scope of the present invention.

Claims

1. A method for communicating commands from a master station (16) to a slave station (12) so that said commands can be authenticated by said slave station (12), said method comprising the steps of:
   (a) having in a master station (16) a code pad (86) common with a code pad (86) in said slave station (12);
   (b) saving said pad at least until a need arises for communicating a command to said slave station (12);
   (c) combining said pad with said command to form a message (89); and
   (d) transmitting said message (89) to said slave station (12).

2. A method as claimed in Claim 1 wherein:
   said step (a) wherein said master station (16) receives (168) a set of pads from said slave station (12);
   said step (b) saves said set of pads (170);
   said step (c) combines a selected one of said pads with said command.

3. A method as claimed in Claim 1 or 2 additionally comprising the step (144) of transmitting a set-up data communication message to said slave station (12), said set-up message being configured to convey an encryption key, said encryption key being transmitted in a form which does not require decryption by said slave station (12).

4. A method for operating a slave station (12) to authenticate commands received from a master station (16) prior to carrying out instructions communicated thereby, said method comprising the steps of:
   (a) randomly generating a pad (154);
   (b) saving said pad and transmitting said pad to said master station (166);
   (c) receiving a data communication message (98);
   (d) recovering a command (98) and a pad value (104) from said message;
   (e) evaluating said recovered pad value to detect correspondence (106) between said recovered pad value and said saved pad; and
   (f) refraining from acting upon said command if said step (e) fails to detect said correspondence (112).

5. A method as claimed in Claim 4 wherein:
   said step (a) generates a set of pads (78);
   said step (b) saves said set of pads and transmits said set of pads;
   said step (e) includes the step of obtaining a selected one (86) of said pads from said saved set of pads, and said step (e) evaluates said recovered pad value using said selected one of said pads to detect said correspondence (106).

6. A method as claimed in Claim 4 or 5 wherein said step (b) includes the step of encrypting said pad using an encryption key so that said pad is transmitted in an encrypted form (160-166).

7. A method as claimed in Claim 4, 5 or 6 additionally comprising the steps of:
   examining said recovered command to determine whether said recovered command is a critical command (100);
   performing said steps (e) (104) and (f) if said recovered command is a critical command; and
   acting upon said recovered command if said recovered command is not a critical command.

8. A method as claimed in Claim 4, 5, 6 or 7 additionally comprising the step of transmitting a failed authentication message (114) to said master station (16) if said step (e) fails to detect said correspondence so that said master station (16) is alerted that said data communication message

was received but was not authenticated.

9. A method for authenticating commands communicated between a master station (16) and a remotely located slave station (12), said method comprising the steps of:

(a) generating, at said slave device, a set of random pads (86);

(b) saving said pads at said slave device and transmitting said pads from said slave device to said master device;

(c) transmitting a message (96) from said master device to said slave device, said message including a command and a selected one of said pads;

(d) receiving said message at said slave device and recovering a pad value therefrom (98);

(e) evaluating said recovered pad value (104) to detect correspondence (106) between said recovered pad value and said selected one of said saved pads; and

(f) refraining from executing said command (112) at said slave device if said step (e) fails to detect said correspondence.

10. A method as claimed in Claim 9 additionally comprising the steps of:

transmitting a set-up data communication message from said master station (16) to said slave station (12) prior to said step (b), said set-up message conveying an encryption key in a form which does not require decryption before use by said slave station (12);

encrypting, at said slave station (12), said set of pad values prior to transmitting them in said step (b), said encrypting step utilizing said encryption key; and

decrypting, at said master station (16), said encrypted pad values, said decrypting step utilizing a secret decryption key that corresponds to said encryption key.

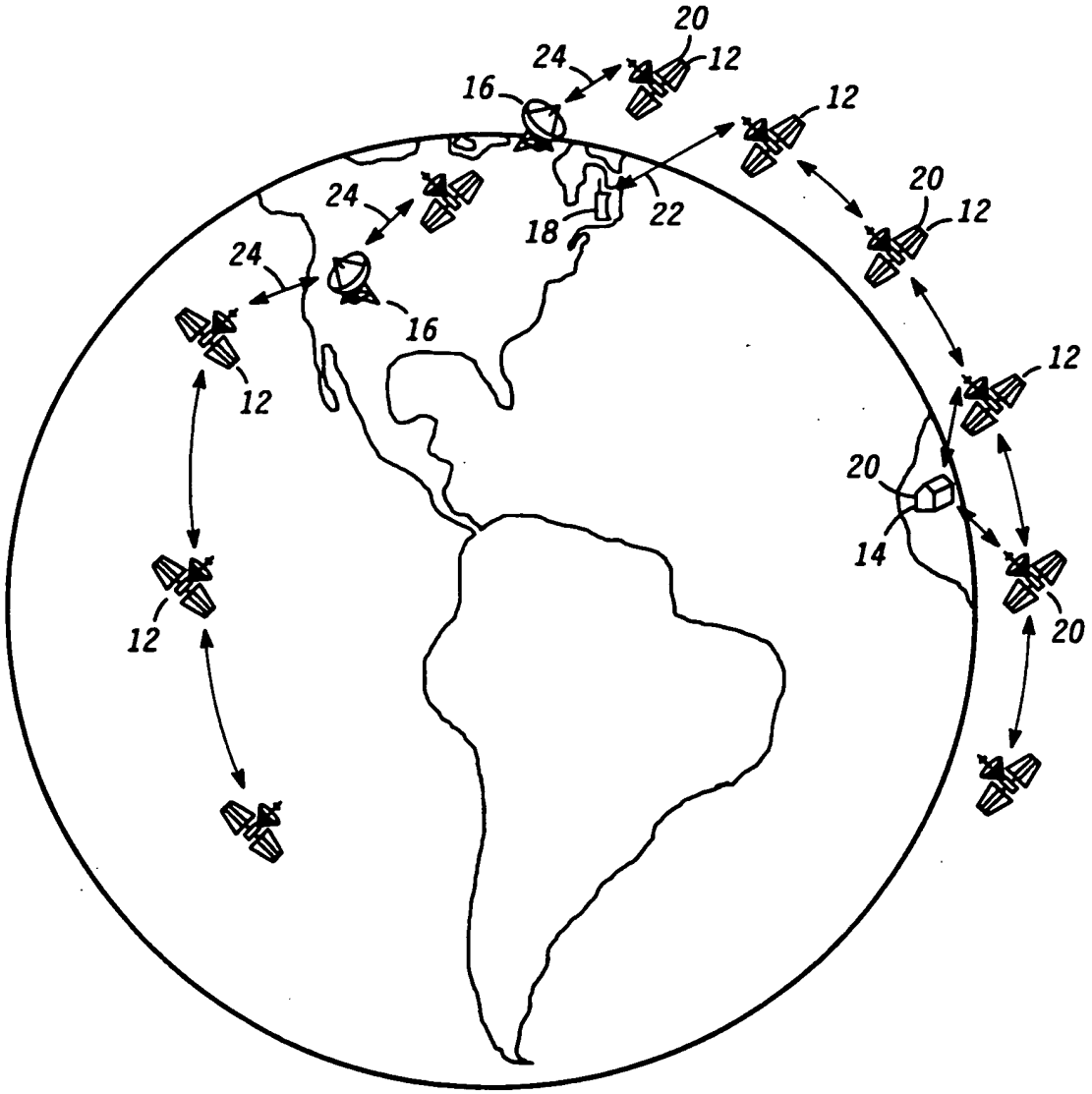


FIG. 1

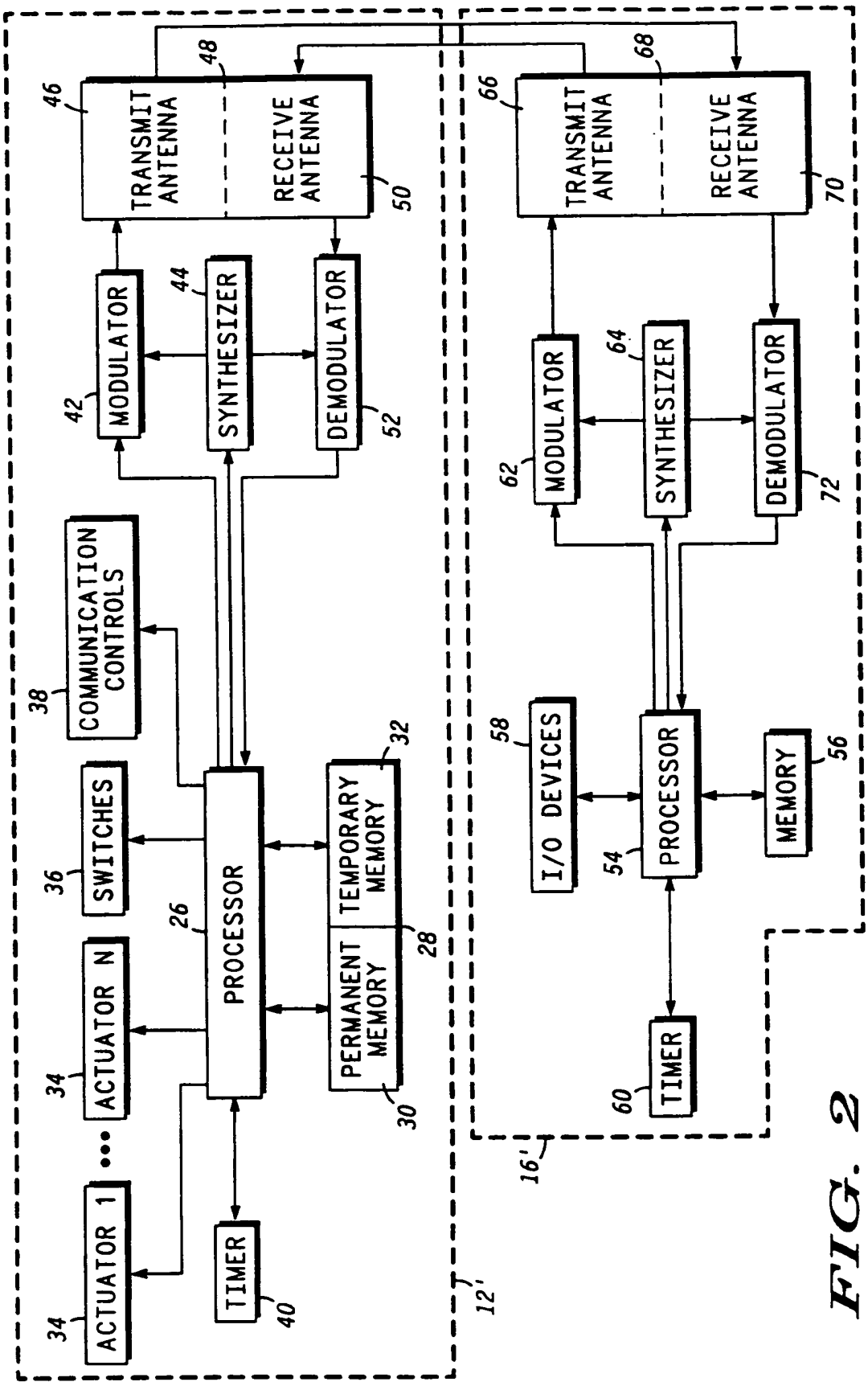

TRANSMIT ANTENNA
RECEIVE ANTENNA
MODULATOR
SYNTHESIZER
DEMODULATOR
COMMUNICATION CONTROLS
SWITCHES
ACTUATOR N
ACTUATOR 1
PROCESSOR
TEMPORARY MEMORY
PERMANENT MEMORY
TIMER
I/O DEVICES
MEMORY
46
48
50
42
44
52
38
36
34
26
32
28
30
40
12'
66
68
70
62
64
72
58
54
56
60
16'

FIG. 2

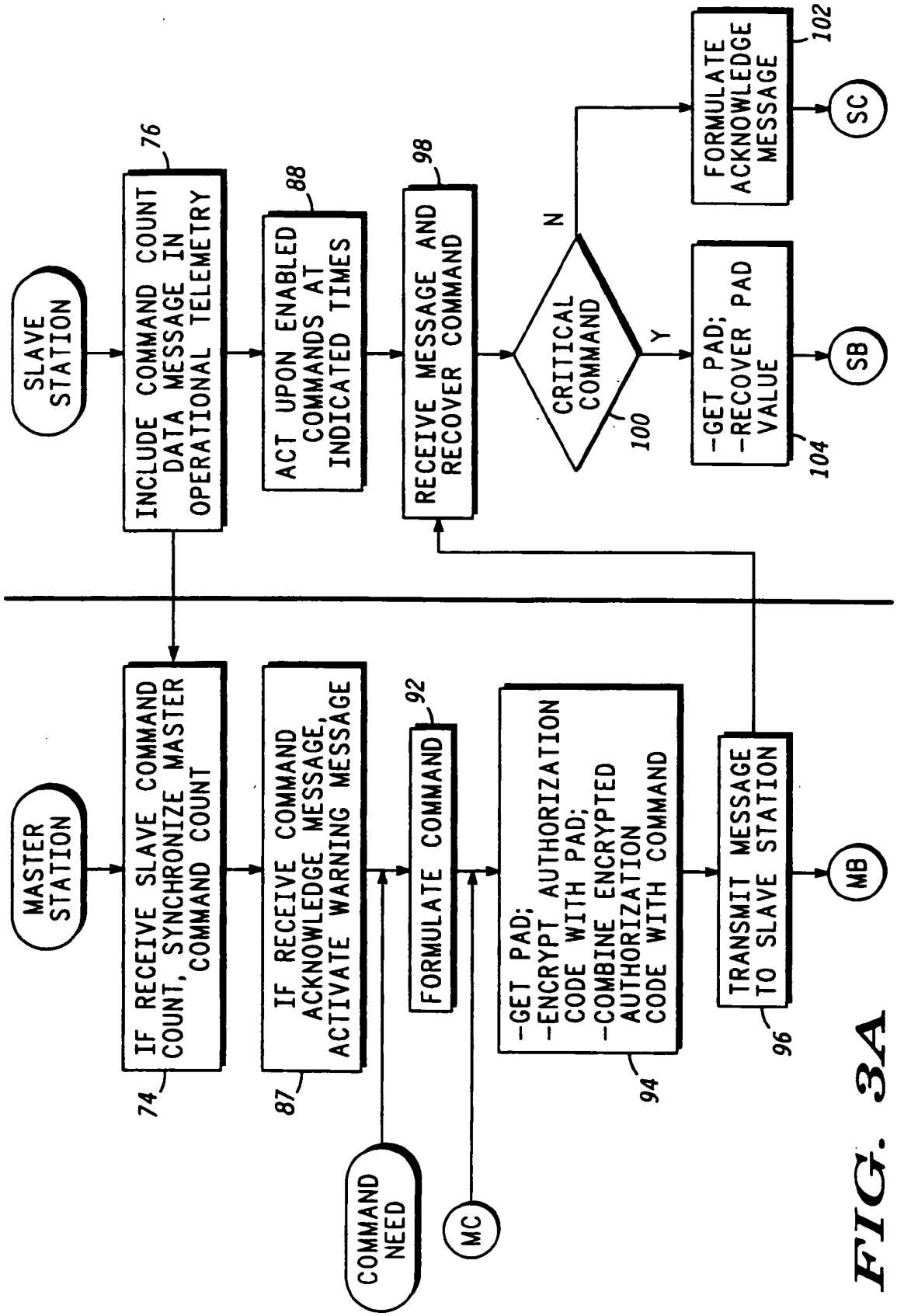

SLAVE STATION
INCLUDE COMMAND COUNT DATA MESSAGE IN OPERATIONAL TELEMETRY
76
ACT UPON ENABLED COMMANDS AT INDICATED TIMES
88
RECEIVE MESSAGE AND RECOVER COMMAND
98
CRITICAL COMMAND
100
N
Y
FORMULATE ACKNOWLEDGE MESSAGE
102
SC
-GET PAD;
-RECOVER PAD VALUE
104
SB
MASTER STATION
IF RECEIVE SLAVE COMMAND COUNT, SYNCHRONIZE MASTER COMMAND COUNT
74
IF RECEIVE COMMAND ACKNOWLEDGE MESSAGE, ACTIVATE WARNING MESSAGE
87
FORMULATE COMMAND
92
-GET PAD;
-ENCRYPT AUTHORIZATION CODE WITH PAD;
-COMBINE ENCRYPTED AUTHORIZATION CODE WITH COMMAND
94
TRANSMIT MESSAGE TO SLAVE STATION
96
MB
COMMAND NEED
MC

FIG. 3A

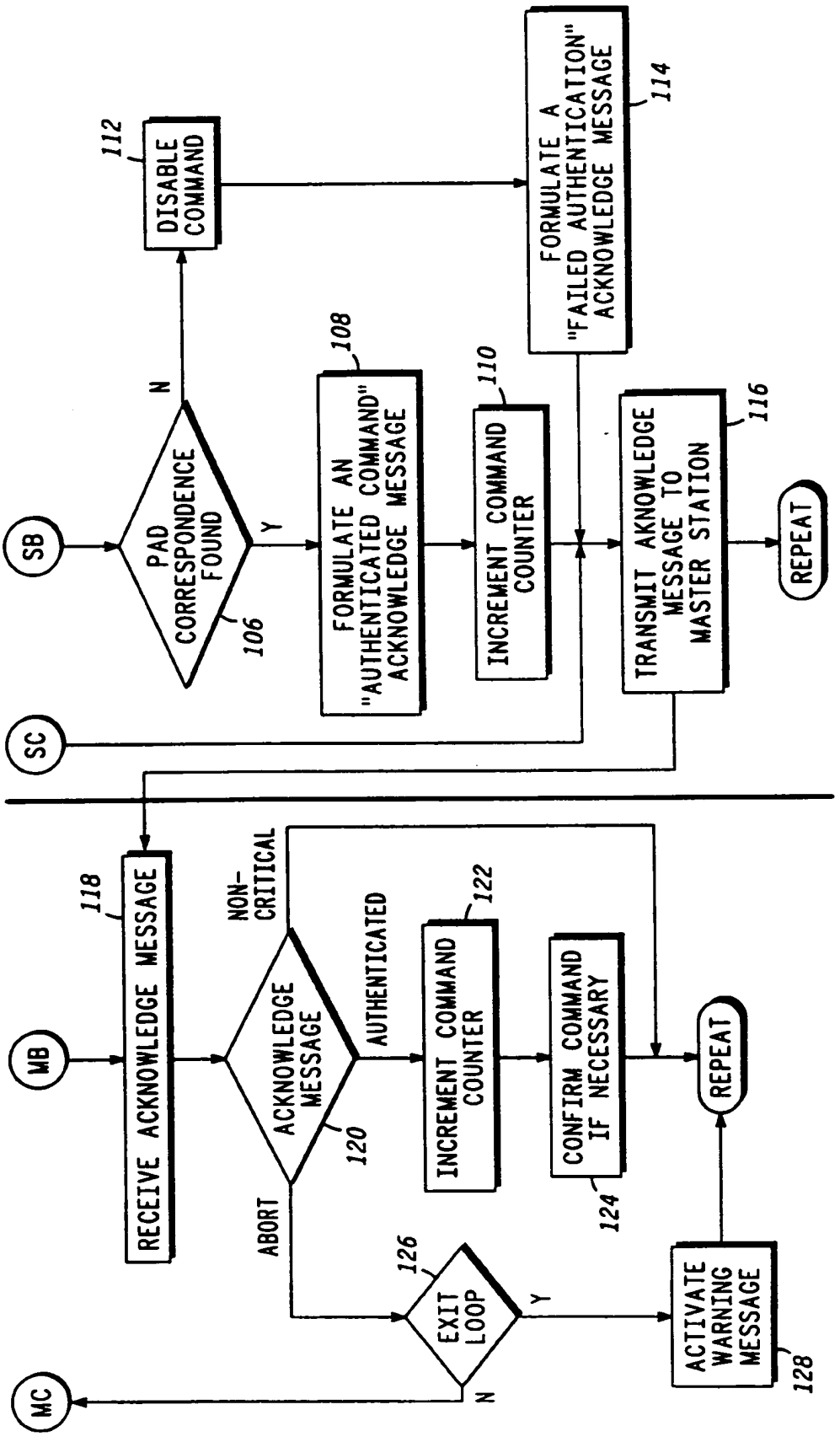

SB
PAD CORRESPONDENCE FOUND
106
N
Y
DISABLE COMMAND
112
FORMULATE AN "AUTHENTICATED COMMAND" ACKNOWLEDGE MESSAGE
108
INCREMENT COMMAND COUNTER
110
FORMULATE A "FAILED AUTHENTICATION" ACKNOWLEDGE MESSAGE
114
TRANSMIT AKNOWLEDGE MESSAGE TO MASTER STATION
116
REPEAT
SC
MB
RECEIVE ACKNOWLEDGE MESSAGE
118
ACKNOWLEDGE MESSAGE
120
NON-CRITICAL
AUTHENTICATED
ABORT
INCREMENT COMMAND COUNTER
122
CONFIRM COMMAND IF NECESSARY
124
REPEAT
EXIT LOOP
126
Y
N
ACTIVATE WARNING MESSAGE
128
MC

FIG. 3B

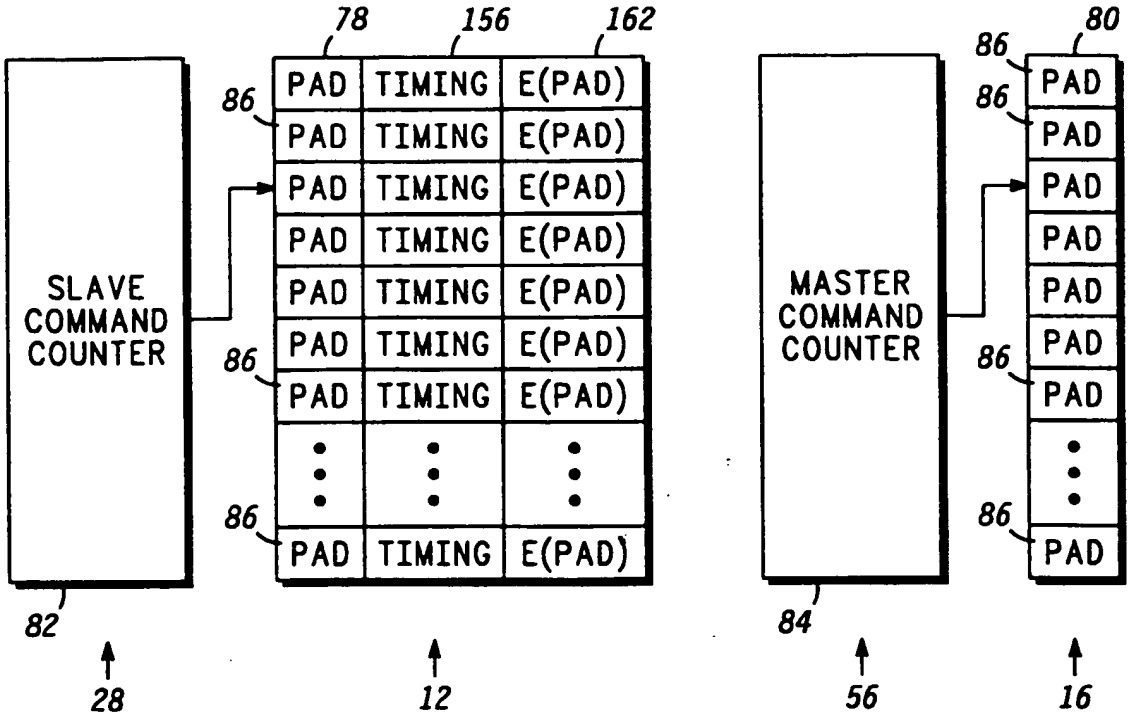


FIG. 4

FIG. 5A

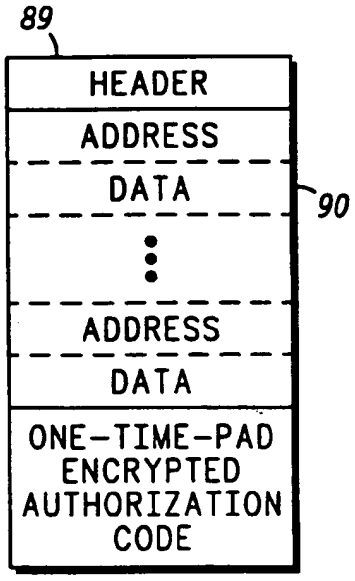
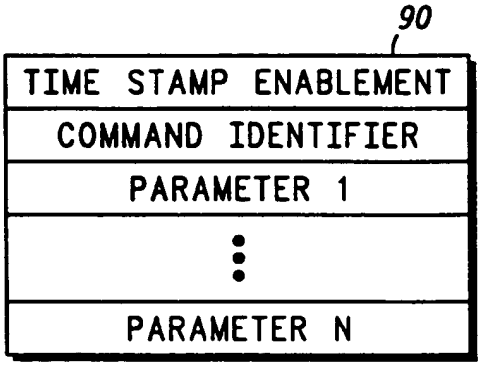


FIG. 5B



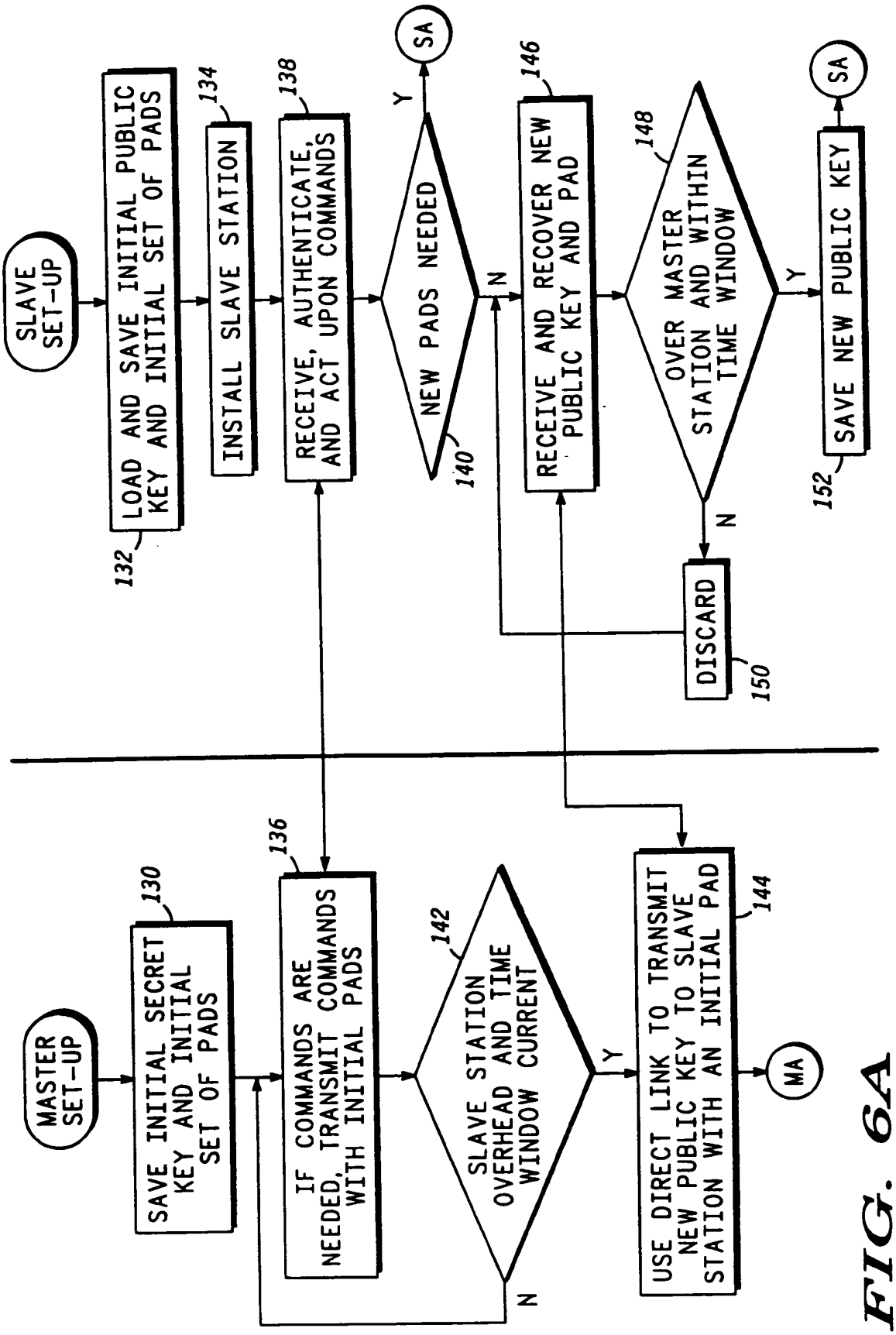

MASTER SET-UP
SAVE INITIAL SECRET KEY AND INITIAL SET OF PADS
130
IF COMMANDS ARE NEEDED, TRANSMIT COMMANDS WITH INITIAL PADS
136
SLAVE STATION OVERHEAD AND TIME WINDOW CURRENT
142
N
Y
USE DIRECT LINK TO TRANSMIT NEW PUBLIC KEY TO SLAVE STATION WITH AN INITIAL PAD
144
MA
SLAVE SET-UP
LOAD AND SAVE INITIAL PUBLIC KEY AND INITIAL SET OF PADS
132
INSTALL SLAVE STATION
134
RECEIVE, AUTHENTICATE, AND ACT UPON COMMANDS
138
NEW PADS NEEDED
140
Y
SA
N
RECEIVE AND RECOVER NEW PUBLIC KEY AND PAD
146
OVER MASTER STATION AND WITHIN TIME WINDOW
148
N
DISCARD
150
Y
SAVE NEW PUBLIC KEY
152
SA

FIG. 6A

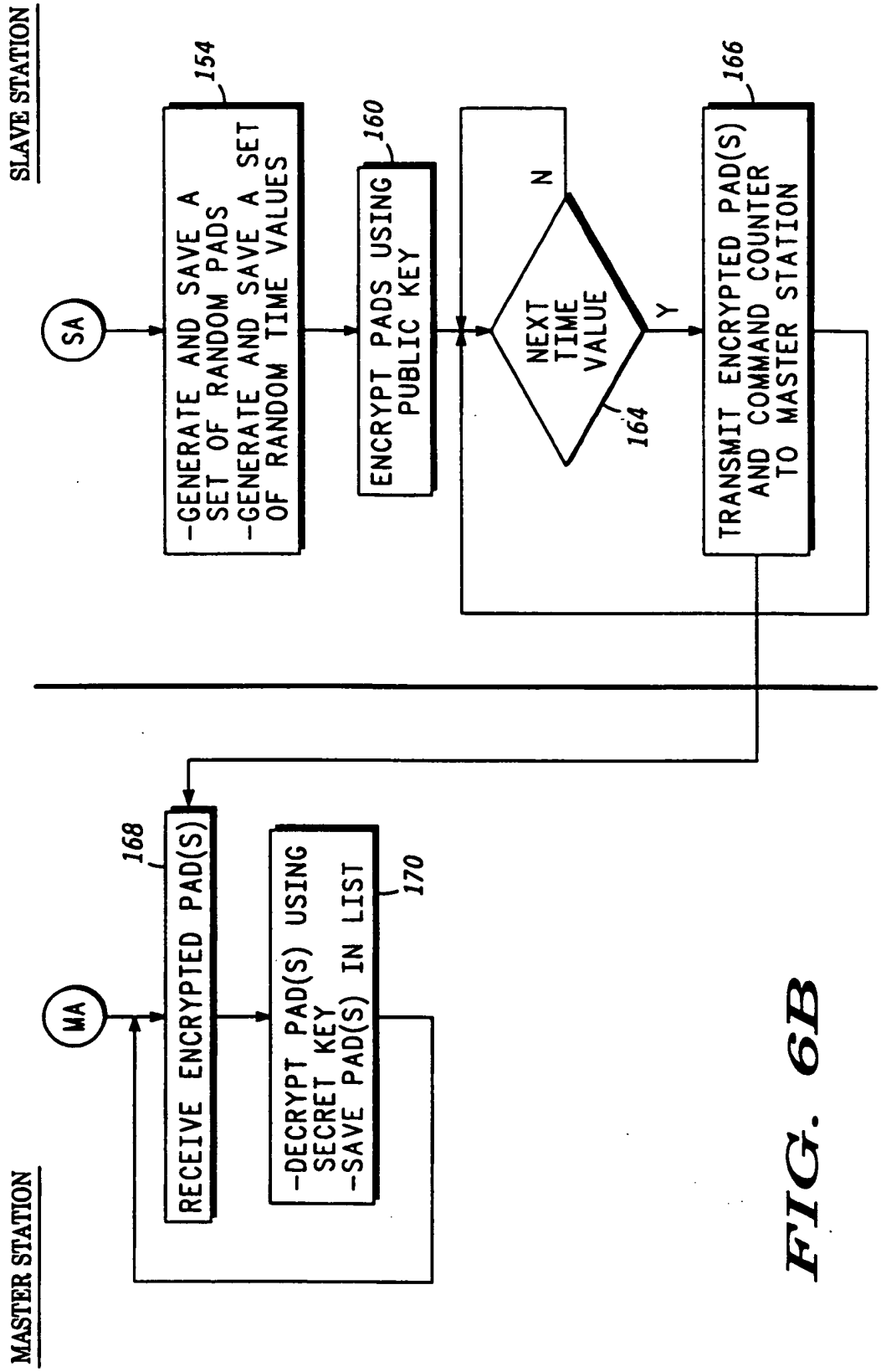

SLAVE STATION
SA
154
-GENERATE AND SAVE A SET OF RANDOM PADS
-GENERATE AND SAVE A SET OF RANDOM TIME VALUES
160
ENCRYPT PADS USING PUBLIC KEY
NEXT TIME VALUE
164
N
Y
166
TRANSMIT ENCRYPTED PAD(S) AND COMMAND COUNTER TO MASTER STATION
MASTER STATION
MA
168
RECEIVE ENCRYPTED PAD(S)
-DECRYPT PAD(S) USING SECRET KEY
-SAVE PAD(S) IN LIST
170

FIG. 6B